July 14, 1952

Memo to: J. DeTurk

From : J. Carr III

Subject: Programming on the Whirlwind Computer

ABSTRACT:

The following memorandum is a reproduction of a paper given by the author at the May meeting of the ACM in Pittsburgh. It should be of interest to both programmers and people interested in building computers because of its presentation of the philosophy of programming at the Massachusetts Institute of Technology, which is in some ways different from that proposed for the MIDAC.

Distribution:

Group 1430
Group 1640

Approved by J. DeTurk

Date July 30

UNCLASSIFIED

Memo to: J. DeTurk

From : J. Carr III

Subject: Programming on the Whirlwind Computer

PROGRESS OF THE WHIRLWIND COMPUTER TOWARDS AN AUTOMATIC PROGRAMMING PROCEDURE

Summary

This paper shall discuss present and proposed uses of subroutines and other pre-tested automatic programs on the Whirlwind computer at the Digital Computer Laboratory, Massachusetts Institute of Technology. At Whirlwind, programming methods are involved not only with a subroutine Library, but also with the use of artificial programmed codes, interpretive routines, automatic assembly schemes, and conversion programs. For this reason, this paper shall attempt both to summarize the use of all these devices and to show the directions that final organization of the machine's program structure may take.

Advantages of Subroutines

The growing use of subroutines on high-speed automatic computing machines requires a re-evaluation of their use and prupose, as viewed in the light of the overall program structure of the machine. Although some hold-outs exist, most people involved in setting up problems for high-speed computing machines admit the usefulness of subroutines. In fact, the subroutine idea is basic to any machine. Any instruction can be considered to be a wired-in subroutine combining electronically many micro-instructions.

The use of a set of instructions, or order code, in a computing machine has, among others, the following advantages.

1) The instructions are automatic in operation and in sequencing to the next instruction.

2) The instructions are pre-tested, and so their procedure of operation does not have to be re-written and checked before each use.

3) No provision has to be made for storage of the results of intermediate steps--partial sums, etc.--in the course of the performance of the instruction.

4) Storage of instructions, upon read-in into the machine, is performed sequentially and automatically, so that no provision has to be made by the programmer in order for the next instruction to go in the succeeding storage location.

5) The use of instructions--wired-in subroutines--saves a great deal of time and effort on the part of the mathematician or programmer, since he has to write only a very short coded instruction to represent a complicated operation.

An obvious goal for the user of subroutines--programmed instructions--would be that they should have all the above advantages of the electronic instructions. In addition they would automatically have the following additional advantages:

6) The subroutines may be changed at will, without interfering with machine hardware, rewiring circuits, shutting down operation, etc., simply by changing the combination of electronic instructions.

7) The subroutines are designed on the basis of actual need by the users of the machine, rather than on the basis of hypothesized future needs by designers who in most cases are not the users of the machine.

It must not be overlooked that subroutines, like the wired-in electronic instructions, suffer from several disadvantages. They often require more machine storage than a one-shot routine (that is, one coded for a single specific purpose). Too, subroutines that are to cover a multitude of eventualities may often require more machine time than similar one-shot routines, just as an addition instruction that deals only with positive numbers could be made to operate faster than one which must handle all numbers irrespective of sign.

Most of the subroutines and subroutine organizations devised up to now have included the advantages 1 - 3 listed above. Perhaps the first general use of such subroutines was at EDSAC, the machine of the Mathematical Laboratories at Cambridge University. However, until recently, very little effort has seemingly been made to write subroutines and organize the programming structure of a machine, so that advantages 4 and 5 would be incorporated into the use of subroutines. Both recent and present efforts in programming at Whirlwind have been to incorporate these two ideas --automatic sequential storage and programmed instruction codes--into the use of sub-